

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEK
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Diplomski studij računarstva

Ugradbeni računalni sustavi

Projektni zadatak

Aparat za kavu

Studenti:

Igor Radić,
Matija Rajčevac,
Luka Perić,
Domagoj Vukadin

Osijek, 2021.

SADRŽAJ

1. UVOD	1
1.1. Zadatak i struktura rada.....	1
2. TEORIJSKI OSVRT NA PROBLEM SUSTAVA ZA PRIPREMU KAVE.....	2
2.1. Teorijski osvrt na problem sustava za pripremu kave	2
2.2. Prijedlog sklopovskog rješenja.....	2
2.3. Prijedlog programskog rješenja.....	3
3. REALIZACIJA SUSTAVA	5
3.1. Korišteni alati i programska okruženja.....	5
3.1.1. STM32F4 Discovery razvojna pločica	5
3.1.2. Atollic TrueStudio	5
3.1.3. ESP8266 01 Wi-Fi modul	6
3.1.4. Arduino IDE.....	6
3.1.5. MIT App Inventor	6
3.1.6. Servo motor Towerpro MG995	7
3.1.7. DS18B20 senzor temperature	7
3.1.8. 2-kanalni relej.....	8
3.1.9. Pretvarač logičkih razina.....	9
3.1.10. Pumpa.....	9
3.2. Sklopovsko rješenje.....	10
3.3. Algoritamsko rješenje.....	15
3.3.1. STM32.....	15
3.3.2. Mobilna aplikacija.....	20
3.3.3. ESP8266 Wi-Fi modul	23
4. TESTIRANJE I REZULTATI	25
4.1. Metode i načini testiranja	25
4.2. Rezultati testiranja.....	25
5. ZAKLJUČAK	31
6. LITERATURA.....	32
7. PRILOG	34
7.1. STM32 programski kod.....	34
7.1.1. Korištene biblioteke i globalne varijable	34
7.1.2. Main funkcija	34
7.1.3. Ostale korištene funkcije.....	35

7.2.	ESP8266 programski kod.....	40
7.2.1.	Korištene biblioteke i globalne varijable	40
7.2.2.	Setup funkcija	40
7.2.3.	Loop funkcija	41
7.3.	Mobilna aplikacija programski blokovi.....	43

1. UVOD

U današnjem društvu konzumacija kave, bilo obične crne kave ili instant kave, sastavni je dio života većine ljudi. Većina kavu konzumira zbog kofeina koji ubrzava rad ljudskog metabolizma i daje osjećaj budnosti, s druge strane, neki ljudi kavu piju iz navike, a neki zbog čistog užitka koji njen okus pruža. Običnu crnu kavu je moguće pripremiti na razne načine, pomoću aparata s filterom, korištenjem moka-posude, u French-press vrču ili u džezvi, dok je za instant kavu dovoljna samo vrela voda. Sam proces pripreme kave otvorio je mnoga radna mjesta unutar ugostiteljski objekata kojima je prva i osnovna zadaća pripremiti kavu za svoje goste. S pojavom novih tehnologija i mogućnosti koje one pružaju zasigurno se u jednom trenutku razmatralo ima li smisla da ljudi i dalje sami sebi pripremaju kavu koju bi za njih to mogao napraviti stroj. Ovaj projektni zadatak predlaže rješenje pripreme kave na način da osoba putem mobilne aplikacije odabere kavu prema svom ukusu i sav trud oko pripreme prepusti automatiziranom sustavu. Takav sustav za cilj ima praviti kavu uz minimalna odstupanja u svakom pojedinom sastojku.

1.1.Zadatak i struktura rada

Zadatak ovog projekta je izraditi sustav koji će automatizirati proces kuhanja kave, te korisniku omogućiti naručivanje kave dok je udaljen od samog sustava. Takav sustav biti će primjenjiv u kućanstvima, te može imati puno prostora za daljnje usavršavanje i nadogradnju. Samim time se može nadograditi do razine profesionalnih aparata te jednog dana postati aparat koji će se koristiti u caffeterijama sa načinom pametnog naručivanja kave pomoću mobilnih uređaja.

U prvim poglavljima pružen je uvod u problematiku, te zadatak ovog rada. Drugo poglavlje daje teorijski osvrt na problem te prijedlozi za sklopovsko i programsko rješenje navedenog problema. Treće poglavlje predstavlja realizaciju sustava, opis korištenih alata i programskih okruženja, te sklopovsko i programsko rješenje problema. U četvrtom poglavlju opisuju se metode testiranja i rezultati testiranja te analiza dobivenih rezultata. Na kraju se nalazi zaključak, korištena literatura i prilozi iz projekta.

2. TEORIJSKI OSVRT NA PROBLEM SUSTAVA ZA PRIPREMU KAVE

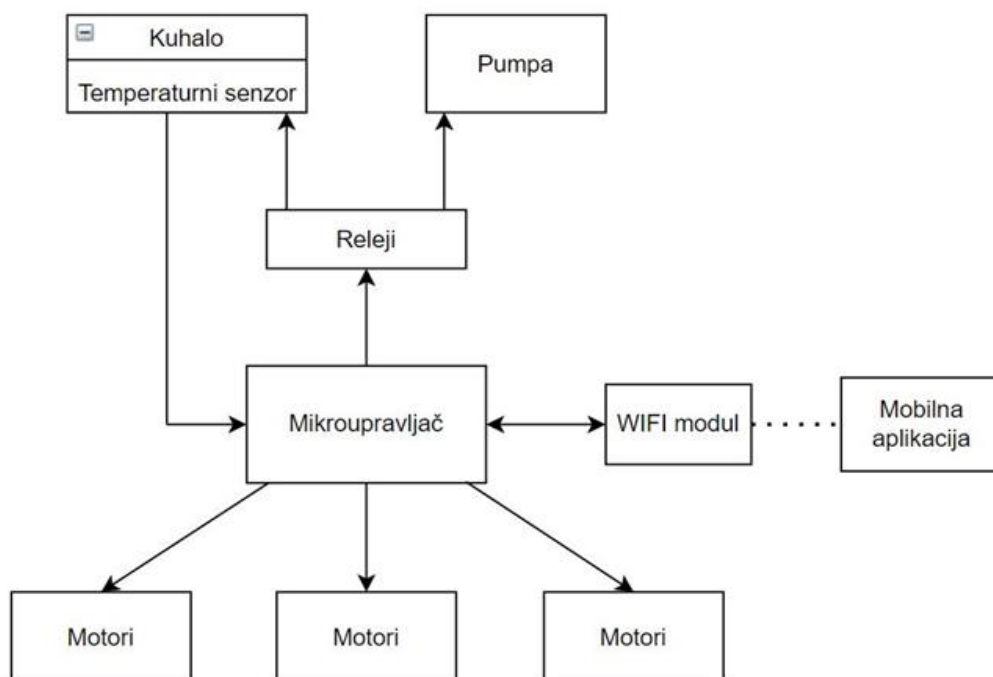
U teoriji postoje razni načini pripreme obične crne kave, a svaki način daje različit okus i posebnu notu koju ljudska osjetila mogu razaznati. Kako bi se ovaj problem donekle razriješio i ograničile moguće komplikacije u ovome projektnom zadatku razmatra se teorija pripreme instant kave u automatiziranom sustavu. Instant kava u zrnju ili prahu miješa se sa vrelom vodom kako bi se kava što bolje rastopila u vodi. Osim kave dodaju se mlijeko u prahu i šećer. Ključan problem jest doziranje određenog sastojka sa što manjim odstupanjima kako bi se svaki puta iznova dobila kava istog okusa. Osim točno doziranih sastojaka, bitna je konstantna količina i temperatura vode u kojoj se sastojci otapaju.

2.1. Teorijski osvrt na problem sustava za pripremu kave

Korisnik će moći odabrati željenu kavu preko mobilne aplikacije. Nakon što korisnik odabere parametre po kojima želi da mu kava bude pripremljena, aplikacija šalje te parametre aparatu za pripremu kave. Aparat na temelju parametara precizno dozira kavu, šećer i mlijeko. Doziranje će biti realizirano otvaranjem donjeg otvora spremnika u obliku lijevka, a sila teža će učiniti da kava, šećer i mlijeko dospiju u šalicu. Vodu će biti potrebno zagrijati do određene temperature, kako bi se njome mogla napraviti kava. Zagrijavanje vode bit će realizirano običnim grijačem za vodu.

2.2. Prijedlog sklopovskog rješenja

Slika 2.1. prikazuje blokovski funkcionalni dijagram prijedloga rješenja prema kojem je sustav hardverski koncipiran. „Mozak“ sustava predstavlja mikroupravljač koji čini glavnu sastavnicu sustava. Na njega su spojeni svi podsustavi koji omogućavaju izvedbu i rad pravljenja kave. Podsustav čini modul koji se sastoji od dva releja. Na mikroupravljač je preko releja spojena pumpa koja pretače zagrijanu vodu iz kuhala u šalicu. Na drugi relej je spojeno kuhalo. U kahalu se nalazi temperaturni senzor. Uloga temperaturnog senzora je obavještavati mikroupravljač o zagrijanosti vode u kahalu. Kada kuhalo zagrije vodu na određenu temperaturu, mikroupravljač isključuje kuhalo. Kako bi se omogućilo povezivanje mobilne aplikacije i središnjeg uređaja postavljen je Wi-Fi modul. Wi-Fi modul omogućava primanje i slanje informacija mikroupravljaču. Ovisno o odabranom tipu kave mikroupravljač aktivira pojedini servo motor kako bi iz ljevaka iscurila određena doza šećera, mlijeka i kave.

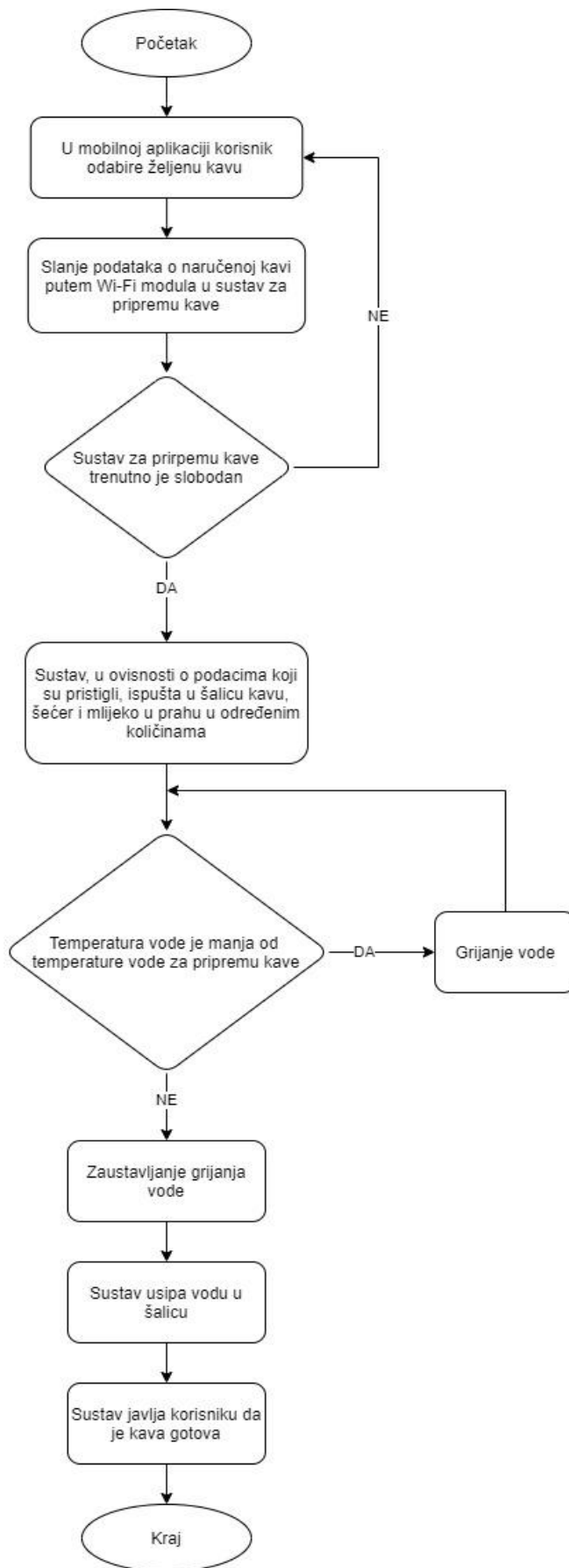


Slika 2.1. Blokovski dijagram sustava za pripremu kave

2.3. Prijedlog programskog rješenja

Kao što je već spomenuto, rješenje za izradu sustava za pripremu kave sastoji se od tri glavna dijela koji međusobno komuniciraju, STM32 mikrokontroler, Wi-Fi modul i mobilna aplikacija. Za programiranje STM32 mikrokontrolera bit će korišteno integrirano razvojno okruženje Atollic TrueStudio, Wi-Fi modul bit će programiran u Arduino IDE-u a mobilna aplikacija korištenjem web aplikacije MIT App Inventor.

Potrebno je dizajnirati grafičko sučelje u mobilnoj aplikaciji koja korisniku omogućava narudžbu željene kave. Mobilni uređaj na kojem se vrši odabir kave pomoću mobilne aplikacije spojen je na Wi-Fi modul sustava za pripremu kave. Podaci o željenoj kavi, nakon što korisnik naruči kavu, šalju se na STM32 mikroupravljač koji potom upravlja procesom kuhanja kave. Algoritam programa na STM32 mikroupravljaču zamišljen je tako da cijelo vrijeme čeka narudžbu kave, a nakon što narudžba dođe kreće izvoditi programski kod koji je vezan za samu izradu kave te za to vrijeme ne može primiti novu narudžbu sve dok ne pripremi kavu. Algoritma kuhanja kave sastoji se od funkcija za dodavanja kave, šećera i mlijeka u prahu u šalicu, provjere temperature vode, zagrijavanja vode do određene temperature te sipanja vode u šalicu. Na kraju procesa kuhanja kave sustav za pripremu kave javlja korisniku, putem mobilne aplikacije, da je kava gotova. Na slici 2.2. prikazan je dijagram tijeka algoritma.



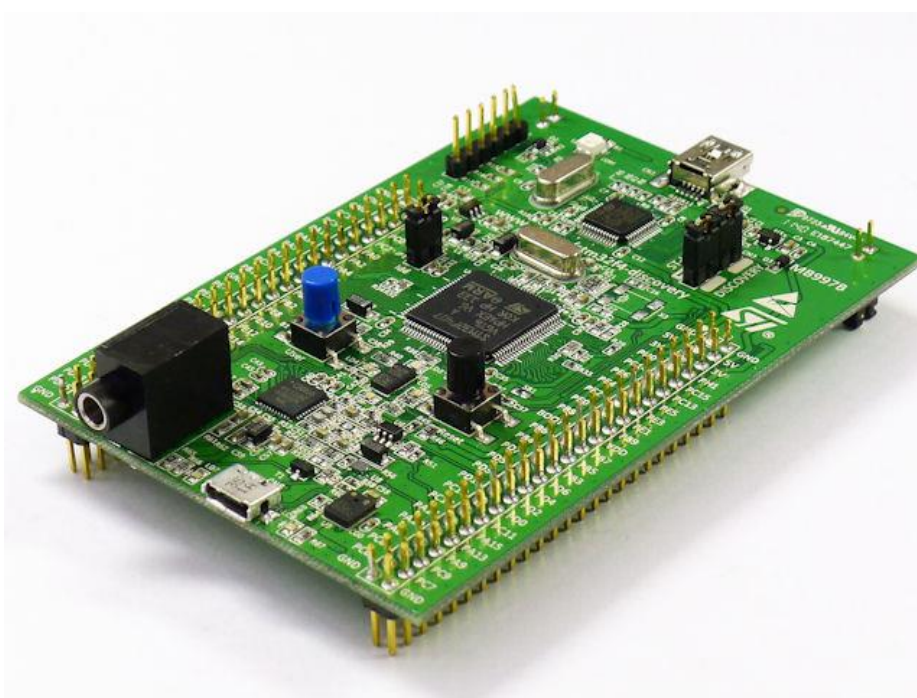
Slika 2.2. Dijagram tijeka algoritma sustava za pripremu kave

3. REALIZACIJA SUSTAVA

3.1. Korišteni alati i programska okruženja

3.1.1. STM32F4 Discovery razvojna pločica

STM32F4 Discovery razvojna pločica sadrži STM32F407VG mikroupravljač koji ima ARM Cortex-M4 32-bitnu jezgru, koja može raditi na frekvenciji do 168 MHz [1]. Radi se o mikroupravljaču visokih performansi koji je uparen sa mnoštvom dodatne periferije. Pločica sadrži 15 komunikacijskih sučelja (I2C, UART, SPI..), 140 ulazno izlaznih pinova koji su pogodni za prototipiranje. Pločica se na računalo spaja putem USB kabla no može biti napajana i sa vanjskog izvora napajanja.



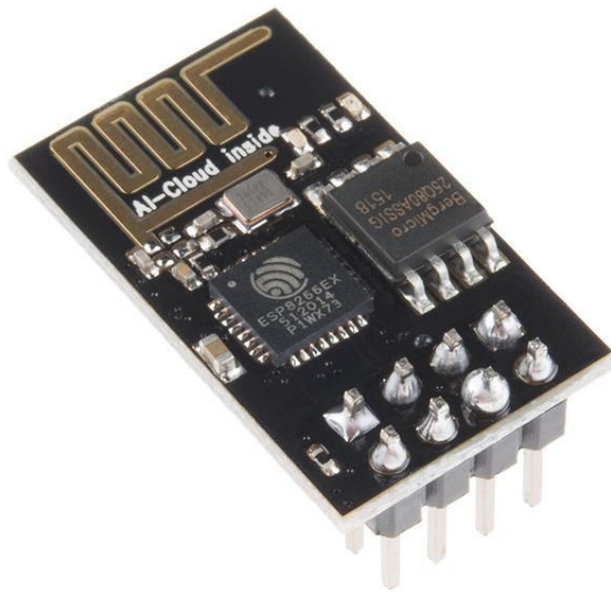
Slika 3.1 STM32F4 Discovery razvojna pločica [2]

3.1.2. Atollic TrueStudio

Atollic TrueStudio razvojno je okruženje koje omogućava programiranje mikroupravljača serije STM32 pomoću C/C++ programskog jezika. Zasnovano je na Eclipse frameworku te podržava programiranje i otklanjanje pogreški putem ST-LINK-a, a ima i mogućnosti analize memorije, prikaza CPU registara i ostale periferije te prikaz pozivanja stoga [1]. Postoji besplatna verzija ovog programskog okruženja koja ne ograničava veličinu programskog koda. Od 2017. godine ne postoji ažuriranje za ovo programsko okruženje, no za potrebe ovog projekta bilo je dovoljno ono što pruža verzija ovog programskog okruženja iz 2017. godine. Atollic TrueStudio također ima ugrađenu podršku za kontrolu verzije programske podrške.

3.1.3. ESP8266 01 Wi-Fi modul

Prema [3], slika 3.2, je Wi-Fi mikročip koji omogućuje ostvarivanje TCP/IP veza sa drugim mikroupravljačima ili uređajima. Ono što ga čini boljim od bluetooth modula jest pristupačnija cijena, manje dimenzije i više mogućnosti. Uz to, osim direktnog pristupa Wi-Fi modulu, moguće je modulu pristupiti putem lokalne mreže ukoliko su modul i uređaj s kojime mu se želi pristupiti spojeni na istu mrežu. Sadrži 32-bitni Tensilica Xtensa LX106 procesor koji radi na frekvenciji 80MHz i 8Mbit-a flash memoriju.



Slika 3.2 ESP8266 01 Wi-Fi modul [4]

3.1.4. Arduino IDE

Prema [5], Arduino IDE je integrirano razvojno okruženje za pisanje programa u C/C++ programskom jeziku koji će se izvoditi na Arduino kompatibilnim mikroupravljačima. Uglavnom se koristi za pisanje koda, kompajliranje istoga i programiranje mikroupravljača s napisanim kodom. Arduino IDE je program otvorenog koda koji se može besplatno preuzeti na službenoj Arduino stranici. Sadrži brojne jednostavne primjere koda za različite mikroupravljače koji uvelike olakšavaju stvaranje programskog koda za specifičnu primjenu.

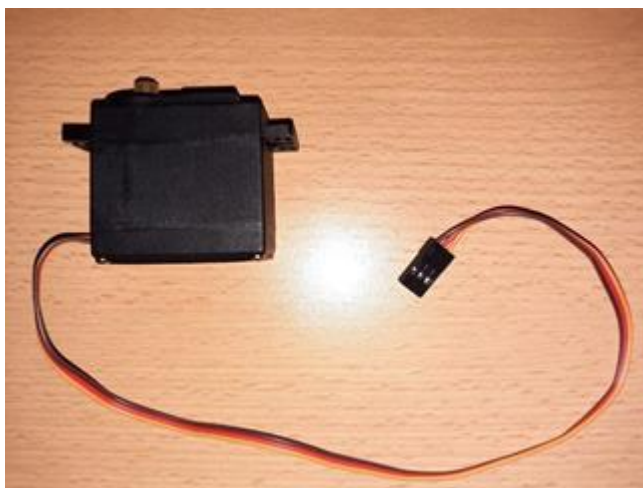
3.1.5. MIT App Inventor

Prema [6], MIT App Inventor je integrirano razvojno okruženje u obliku web aplikacije koje je omogućio Google. Trenutno je za održavanje MIT App Inventora zadužen Massachusetts Institute

of Technology (MIT). Ovo razvojno okruženje omogućuje brz i jednostavan razvoj mobilnih aplikacija za Android i iOS operacijske sustave. Zasniva se na grafičkom sučelju sa *drag and drop* elementima koji se povezuju u logičku cjelinu koja izvršava neki zadatak. Primjer funkcijskih blokova za ovaj projektni zadatak nalazi se u poglavlju 7.3.

3.1.6. Servo motor Towerpro MG995

MG995 je snažni servo motor koji u ovom projektnom zadatku služi za zakretanje ručice koja je učvršćena na osovinu motora radi ispadanja određene krutine iz ljevaka. Za napajanje i upravljanje koristi napon od 4.8 V do 7.2 V i ima brzinu okretaja od 0.20s/60 stupnjeva (pri 4.8V) [7]. Motor može okretati osovinu za kut od 360 stupnjeva. Servo motor u sebi sadrži jedan potencijometar, DC motor i malu pločicu. Kako se otpor na potencijometru mijenja pločica upravlja radom DC motora. Servo motori kontroliraju se pomoću PWM signala. Ovisno o duljini pulsa koji prolazi kroz kontrolnu žicu osovinu motora se okreće za željeni kut.



Slika 3.3. Servo motor MG995

3.1.7. DS18B20 senzor temperature

DS18B20 je vodootporni senzor temperature. Na njega je moguće mjeriti temperaturu na 127 mjesta samo sa jednom žicom (jedan pin). Mjerni raspon mu je između -55°C i $+125^{\circ}\text{C}$ i ima pogrešku mjerenja $\pm 0,5^{\circ}\text{C}$ [8]. Dužina digitalne riječi koju senzor može poslati na mikroupravljač može se postaviti na 9, 10, 11 ili 12 bita. Napon napajanja je između 3.3 V i 5V. Koristi *One-wire* protokol. Koristi se u industriji i temperaturno osjetljivim sustavima.



Slika 3.4. DS18B20 senzor temperature [9]

3.1.8. 2-kanalni relej

Ovaj modul se u ovom projektnom zadatku koristi za kontrolirano uključivanje i isključivanje rada pumpe i kuhala. Relej je elektromagnetski prekidač koji služi za kontroliranje rada trošila koja imaju veću struju ili napon od sustava koji treba upravljati trošilima [10]. Struja koja je potrebna za aktivaciju je 20 mA, napon zavojnice je 5V. Relej u sebi sadrži zavojnicu koja ovisno o protoku struje pali ili gasi trošilo. Relej se sastoji od 5 pinova. COM je pin gdje prolazi struja mreže, a NC(normally closed) i NO(normally open) su pinovi koji ovise o tome hoće li se trošilo paliti ili gasiti. Ostala dva pina služe za napajanje zavojnice koja se nalazi između njih. Ovaj modul sadrži dva takva releja što omogućuje kontroliranje dva trošila.



Slika 3.5. 2-kanalni relej modul [11]

3.1.9. Pretvarač logičkih razina

Ako je potrebno neki signal kojemu je logička jedinica na 5V pretvoriti u logičku jedinicu koja je na 3.3V (ili obrnuto) potreban je pretvarač logičkih razina. Prema [12], to je vrlo jednostavan sklop koji na sebi ima *level-shifting* električni krug za svaki od 4 pina koji služi za povećavanje ili smanjivanje napona. Za to se koristi N-kanalni MOSFET i par *pull-up* otpornika.



Slika 3.6. Logic level konverter [13]

3.1.10. Pumpa

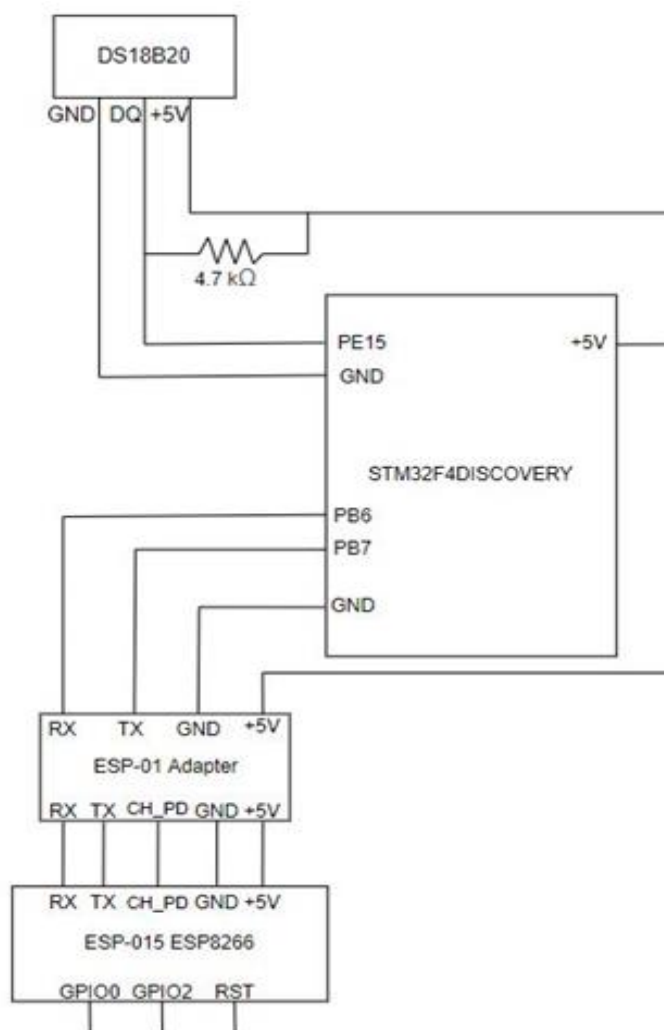
Pumpa sadrži jedan DC motor koji radi na 12V. Pumpa može pumpati vodu zagrijanu i do 100 °C što je za ovaj projekt bilo vrlo važno s obzirom da sustav za pripremu kave mora moći pumpati vodu sa kojom će se pripremati kava. Pumpa je veličine 8.4 x 2.4 cm i ima protok 1.3-1.7 L/min [14].



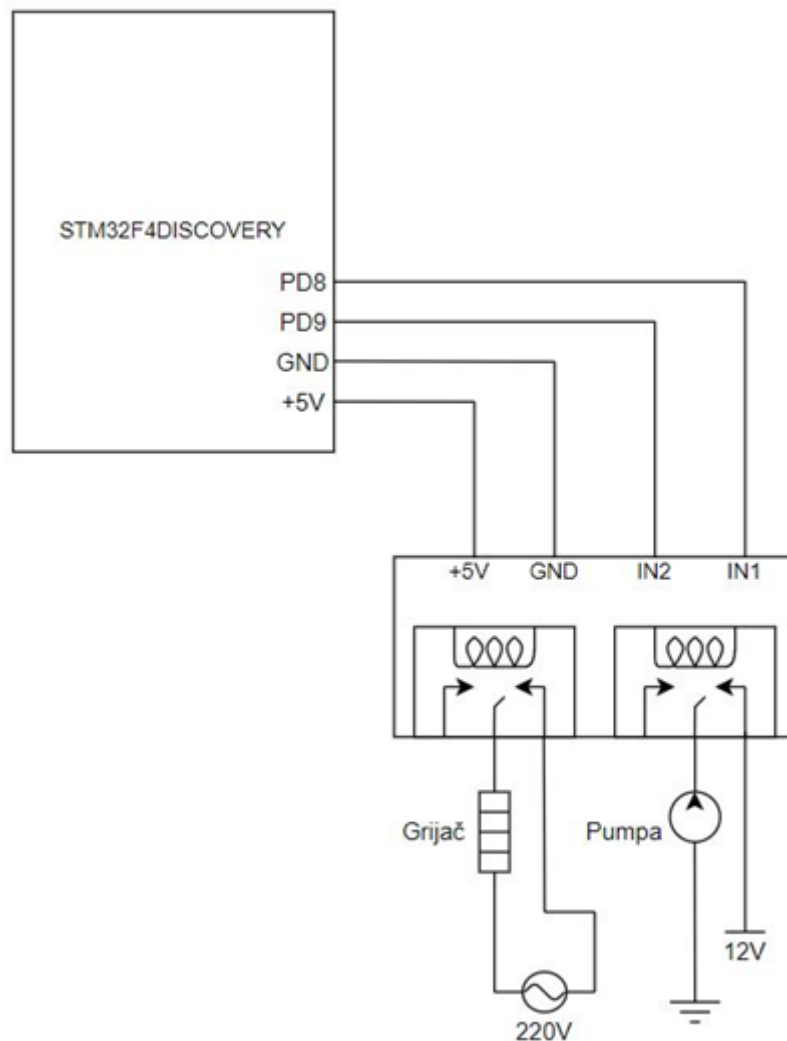
Slika 3.7. Pumpa za vodu

3.2. Sklopovsko rješenje

Povezivanje temperaturnog senzora DS18B20 i STM32F4 Discovery pločice vrlo je jednostavno. +5V žica temperaturnog senzora spojena je na napajanje mikroupravljača od +5 V, a GND žica na GND pin mikroupravljača. Podatkovna žica (DQ) spojena je na digitalni pin PE15 mikroupravljača. Između podatkovne žice i +5 V potrebno je dodati *pull up* otpornik. Otpornik ima vrijednost 4.7 k Ω . Kako bi bilo lakše realizirati komunikaciju Wi-Fi modula i mikroupravljača potreban je ESP-01 adapter. Uzemljenje adaptera se spaja na uzemljenje sustava, kao i napajanje od +5 V, RX pin spaja se na PB6 pin mikroupravljača, a TX na pin. ESP-01 adapter i ESP-015 ESP8266 modul spajaju se pomoću RX, TX, CH_PD, GND i +5 V pinova, čime se na kraju omogućava komunikacija između mobilne aplikacije, Wi-Fi modula i STM32F4 Discovery pločice.

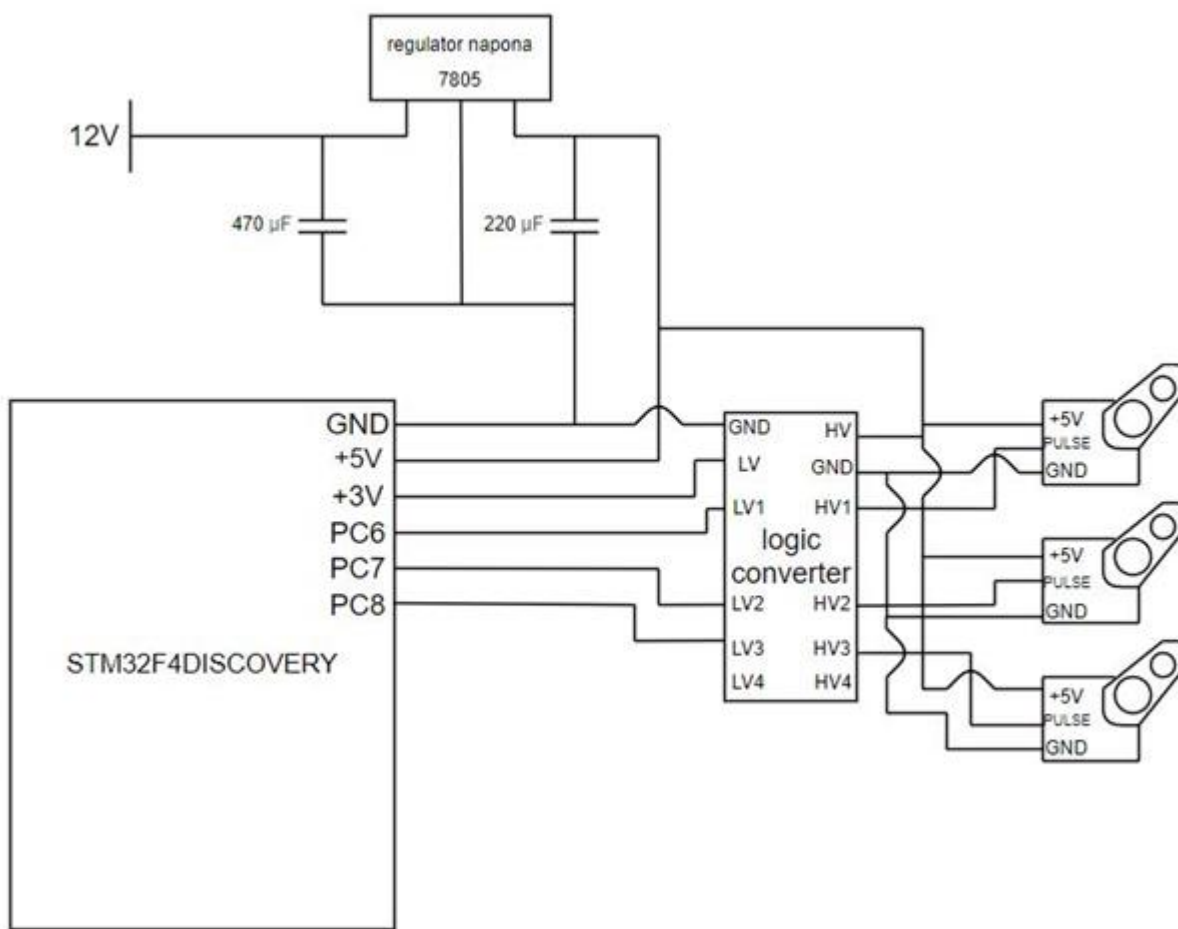


Slika 3.8. Električna shema spajanja Wi-Fi modula i temperaturnog senzora sa STM32F4 Discovery pločicom



Slika 3.9. Električna shema spajanja pumpe i kuhala na STM

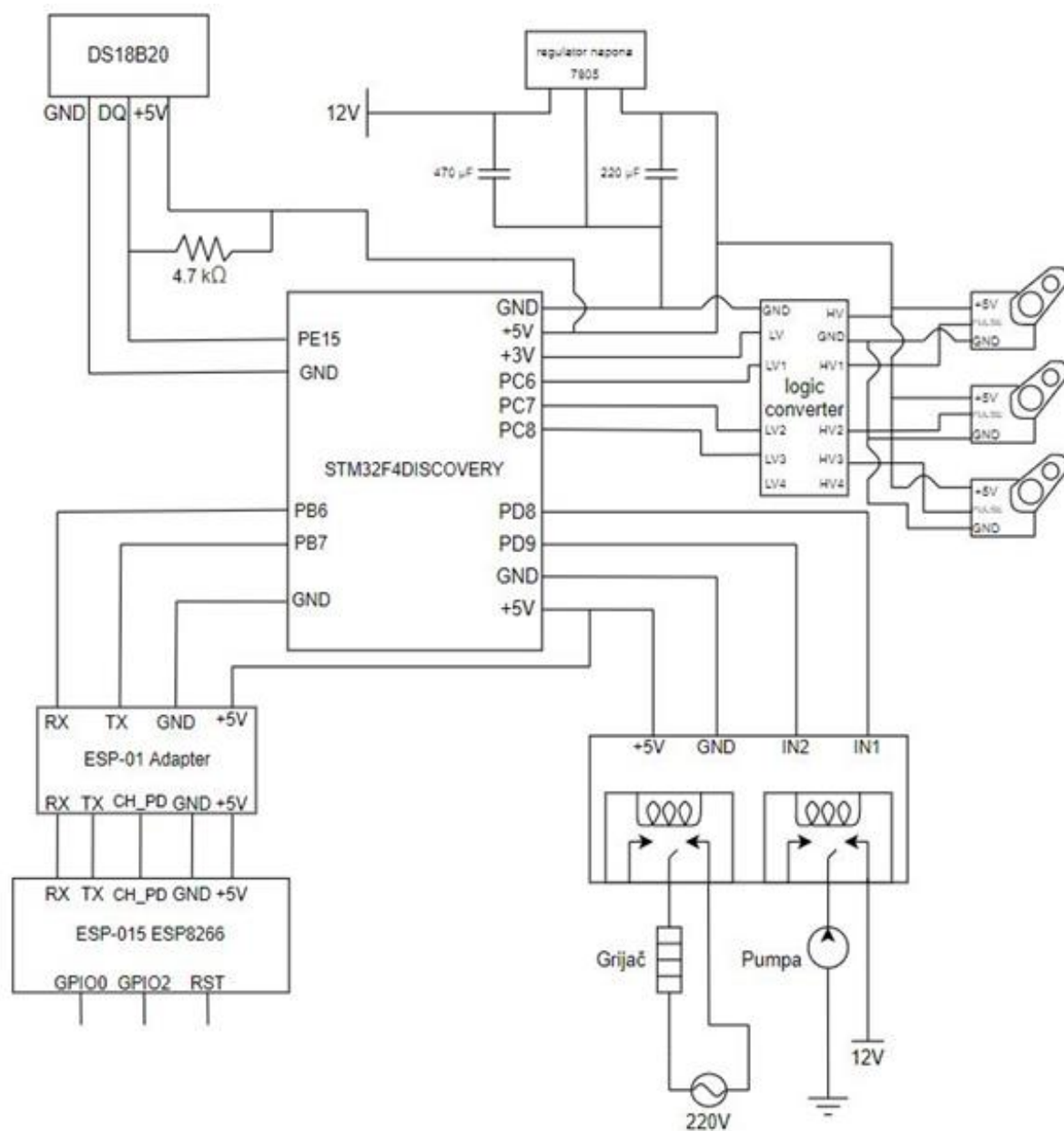
Za zagrijavanje vode u spremniku potreban je grijač. Jedan vod grijača spojen na relej modul koji će zatvarati i otvarati strujni krug grijača. Pumpa služi za punjenje šalice zagrijanom vodom iz spremnika. Jedan ulaz releja spojen je na 12 V, a na zajednički ulaz spojena je pumpa, uzemljenje pumpe spojeno je na uzemljenje izvora napajanja pumpe. Relej modul se s mikroupravljačem spaja pomoću +5 V, GND, IN1 i IN2 pinova. Pin IN1 spojen je na digitalni pin PD8 mikroupravljača dok je IN2 spojen na pin PD9. Prikaz spajanja grijača i pumpe sa relej modulom te spajanja modula sa mikroupravljačem dan je slikom 3.9.



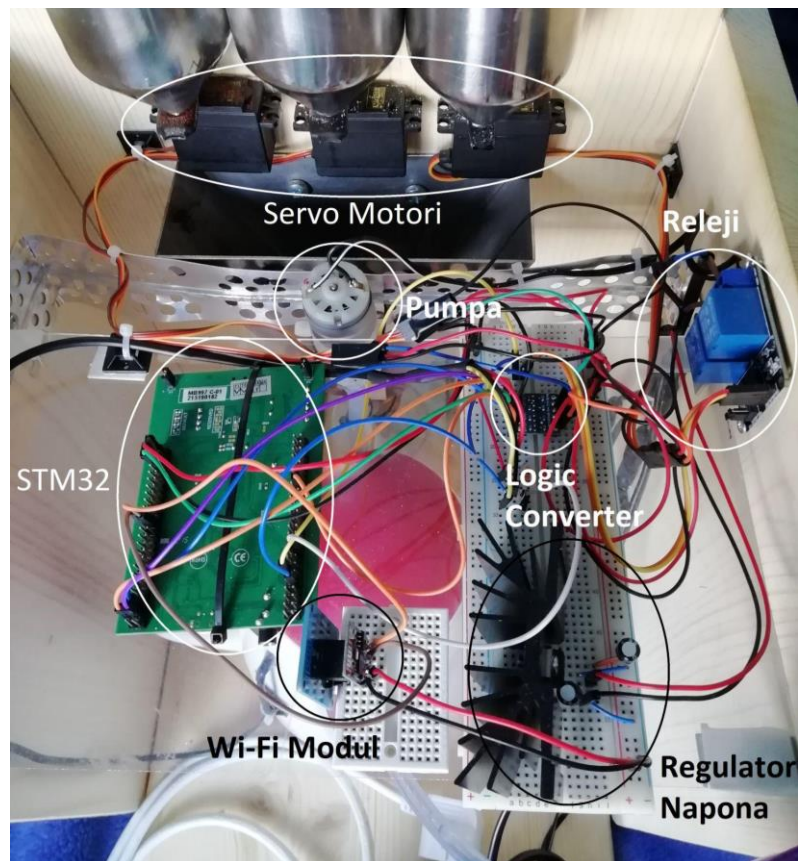
Slika 3.10. Električna shema spajanja motora na STM32 pločicu

Da bi STM32 mikroupravljač mogao ispravno manipulirati servo motorima potreban je pretvarač logičkih razina zbog toga što logičke razine STM32 pločice ne odgovaraju logičkim razinama servo motora. Pinovi PC6, PC7 i PC8 mikroupravljača spajaju na LV1, LV2 i LV3 pinove koji predstavljaju ulaze uređaja sa nižom razinom logičkih stanja (oni kojima je logička jedinica do 3.3V), Osim njih iz mikroupravljača je potrebno spojiti i pinove +3V, +5V i GND na pinove LV, HV i GND pretvarača logičkih razina. +5 V i GND servo motora spojeni su na napajanje i uzemljenje sustava. Treća žica servo motora služi za upravljanje servo motorom pomoću PWM signala. Žice za upravljanje servo motora spojene su na HV1, HV2 i HV3 pinove pretvarača logičkih razina koji predstavljaju izlaze pretvarača za uređaje čija je logička jedinica oko 5V. Jedino dostupno napajanje za ovaj projekt bilo je napajanje od 12V (1.5 A) dok je za napajanje sustava za pripremu kave koje se obrađuje u ovom projektnom zadatku potrebno 5V, da bi se napon od 12 V spustio na potrebnih 5 V potreban je regulator napona 7805 te dva kondenzatora (470 µF i 220 µF). Oni su spojeni prema slici 3.10.

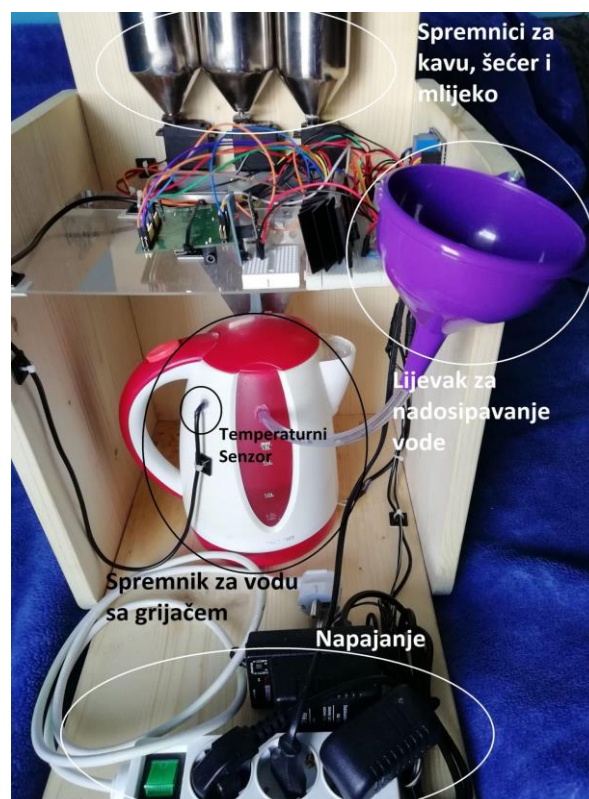
Električna shema cijelog sklopa dana je slikom 3.11.



Slika 3.11. Električna shema cijelog sklopa



Slika 3.12. Prikaz unutrašnjosti makete sustava za pripremu kave



Slika 3.13. Prikaz pozadine makete sustava za pripremu kave



Slika 3.14. Prikaz makete sustava za pripremu kave

3.3. Algoritamsko rješenje

U ovome poglavlju nalazi se objašnjenje algoritma koji koristi sustav za pripremu kave. Budući da se algoritamsko rješenje sustava ostvaruje pomoću tri osnovne komponente (STM32, ESP8266 i mobilna aplikacija), u nastavku se nalazi objašnjenje pojedinog dijela algoritma za svaku komponentu.

3.3.1. STM32

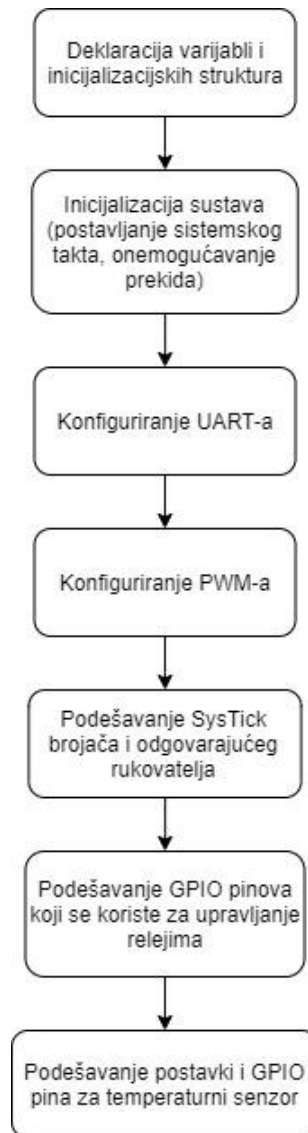
Na samom početku programskog koda za STM32, vrši se inicijalizacija sustava za pravljenje kave. Tu spada deklaracija globalnih i lokalnih varijabli, deklaracija inicijalizacijskih struktura, postavljanje sistemskog takta, onemogućavanje svih prekida, konfiguracija UART-a i PWM-a,

SysTick brojača i odgovarajućeg rukovatelja, uključivanje takta za portove kao i podešavanje inicijalizacijske strukture pomoću koje se rukuje sa temperaturnim senzorom.

Konfiguracija GPIO pinova koji se koriste za upravljanje periferijom vrši se na način da se uključi takt za određeni GPIO port, potom se deklarira inicijalizacijska struktura za GPIO nakon čega je moguće podešavanje određenih postavki GPIO pinova. Podešavanje UART-a i PWM-a također započinje uključivanjem takta za UART odnosno PWM, te deklaracijom potrebnih inicijalizacijskih struktura pomoću kojih je moguće podešavanje. Potrebno je još povezati određene pinove sa UART-om i PWM-om.

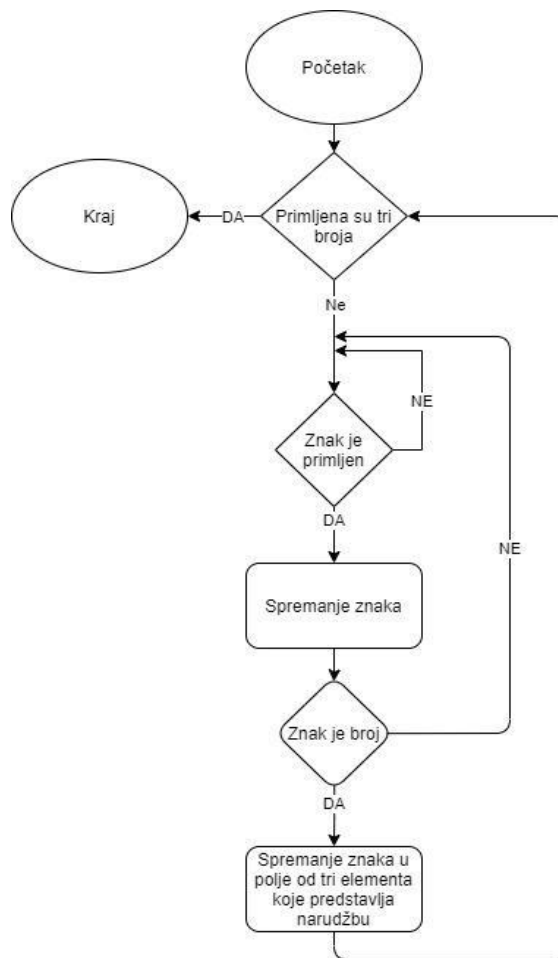
Systick brojač te odgovarajući rukovatelj podešavaju se na način da se prekid generira svake milisekunde kako bi se kasnije ta mogućnost mogla koristiti za *Delay* funkciju koja pauzira izvođenje programa na određeno vrijeme.

Za podešavanje i rukovanje temperaturnim senzorom korištene su biblioteke `tm_stm32f4_onewire.h` [15] i `tm_stm32f4_ds18b20.h` [16]. Na slici 3.15. prikazan je dijagram tijeka inicijalizacije sustava za pripremu kave.



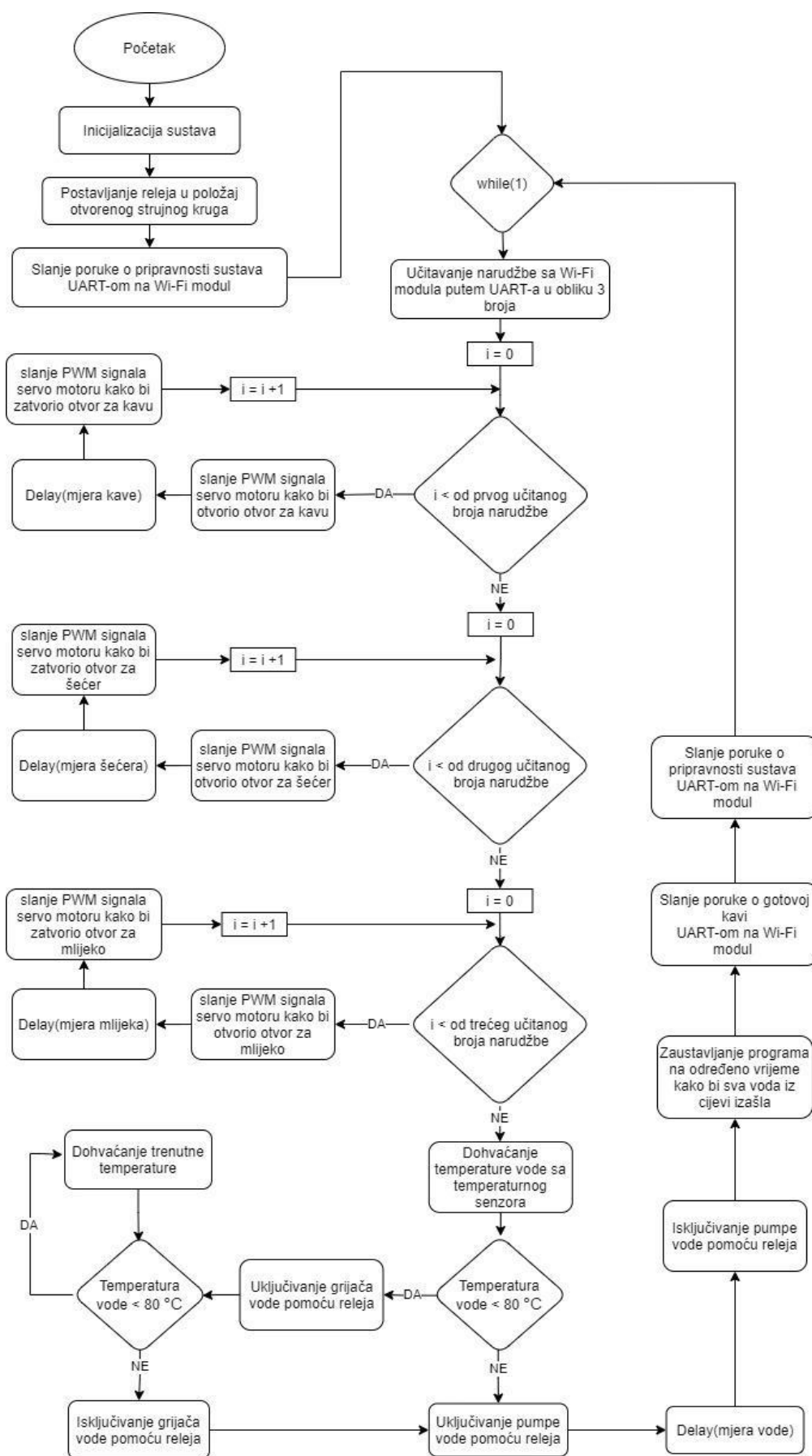
Slika 3.15. Tijek inicijalizacije sustava

Nakon što je sustav inicijaliziran, potrebno je postaviti releje u način otvorenog strujnog kruga kako bi se na početku programa pumpa i grijač za vodu isključili. Potom program ulazi u beskonačnu petlju, u beskonačnoj petlji prvo se učitava narudžba sa Wi-Fi modula putem UART-a. Na slici 3.16. prikazan je tijek algoritma za učitavanje narudžbe.



Slika 3.16. Tijek algoritma za dohvaćanje narudžbe putem UART-a

Narudžba je u programskom kodu predstavljena poljem od tri elementa, prvi element polja predstavlja jačinu kave odnosno broj koliko puta će se ispustiti mjera kave iz spremnika sa kavom, drugi i treći element analogno predstavljaju jačinu šećera odnosno mlijeka. Nakon što je narudžba primljena, pomoću servo motora kontroliranih PWM-signalom frekvencije 50 Hz, otvore se otvori spremnika za kavu, šećer i mlijeko onoliko puta koliko je zadano narudžbom. Potom slijedi dohvaćanje temperature vode u spremniku, ukoliko je ona manja od 80 °C pomoću releja zatvara se strujni krug grijača vode te grijač grije vodu dok ona ne dosegne željenu temperaturu. Ukoliko je temperatura dovoljno visoka prelazi se na uključivanje pumpe za vodu te se program zaustavlja na određeno vrijeme koje je potrebno da pumpa pretoči dovoljno zagrijane vode u šalicu. Pumpa se nakon toga isključuje. Program se ponovno zaustavlja na određeno vrijeme kako bi sve voda iz cijevi pumpe izašla u šalicu. STM32 potom Wi-Fi modulu šalje dvije poruke, prva poruka je poruka koja označava kraj pripremljene kave. Druga poruka označava pripravnost sustava za pripremu kave što znači da je moguće naručiti novu kavu.



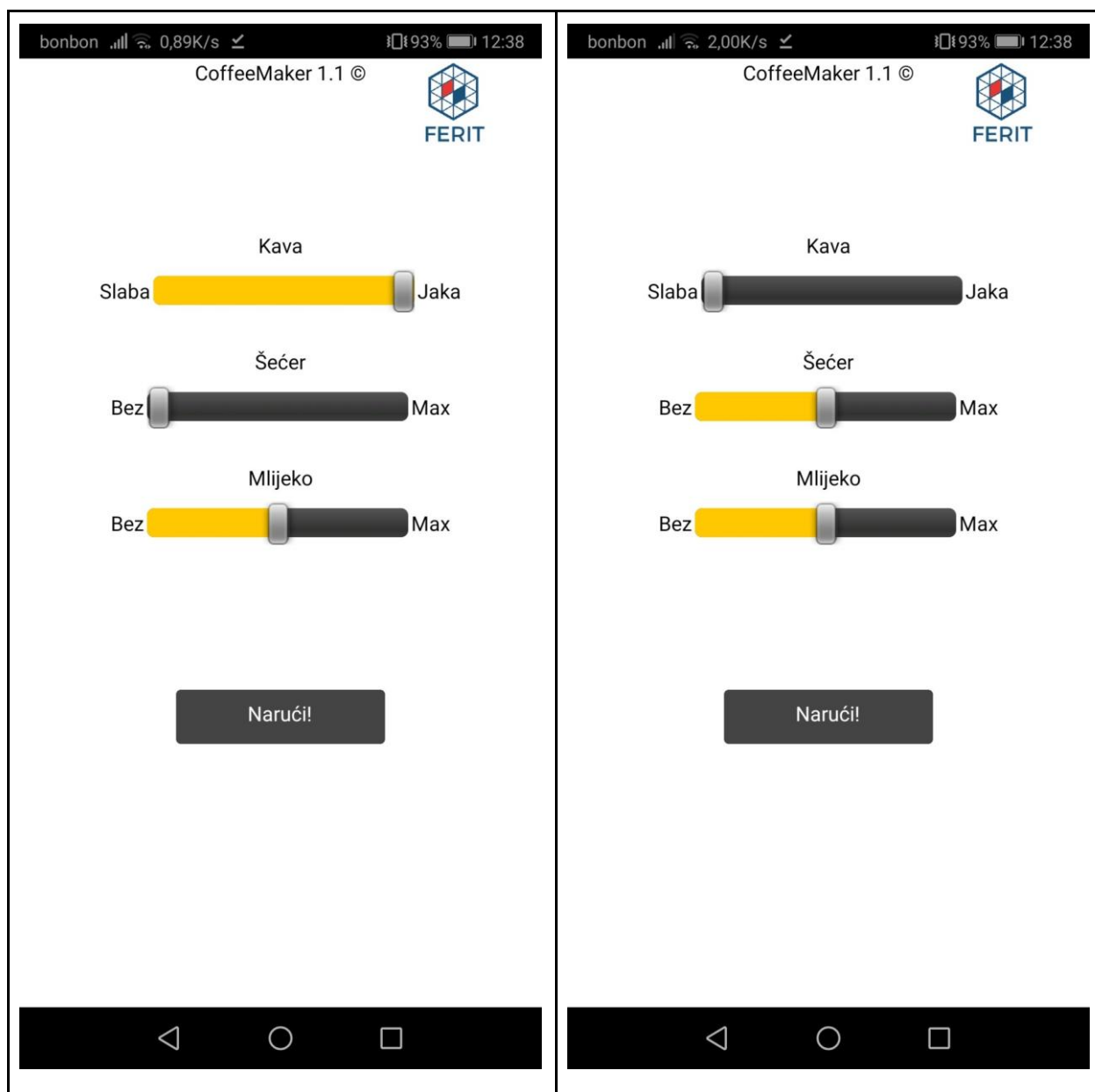
Slika 3.17. Dijagram toka algoritma za STM32 sustava za pripremu kave

3.3.2. Mobilna aplikacija

Sučelje koje omogućuje korisniku narudžbu određene kave ostvareno je u vidu mobilne aplikacije. Cilj aplikacije je jednostavno i intuitivno sučelje čija je zadaća proslijediti parametre kave, koje je korisnik odabrao, putem bežične veze na STM32 pločicu. Korisniku je moguć izbor različitih količina kave, šećera i mlijeka, slika 3.18.

Korisnik ima mogućnost odabrati sljedeće razine:

- Kava:
 - Slaba
 - Srednja
 - Jaka
- Šećer:
 - Bez
 - Jedna doza
 - Dvije doze
- Mlijeko:
 - Bez
 - Jedna doza
 - Dvije doze

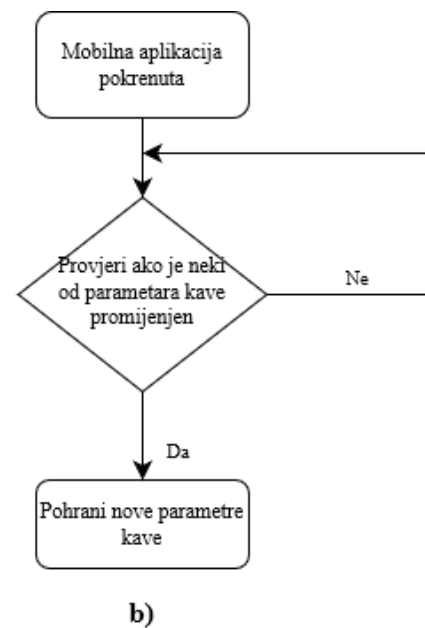
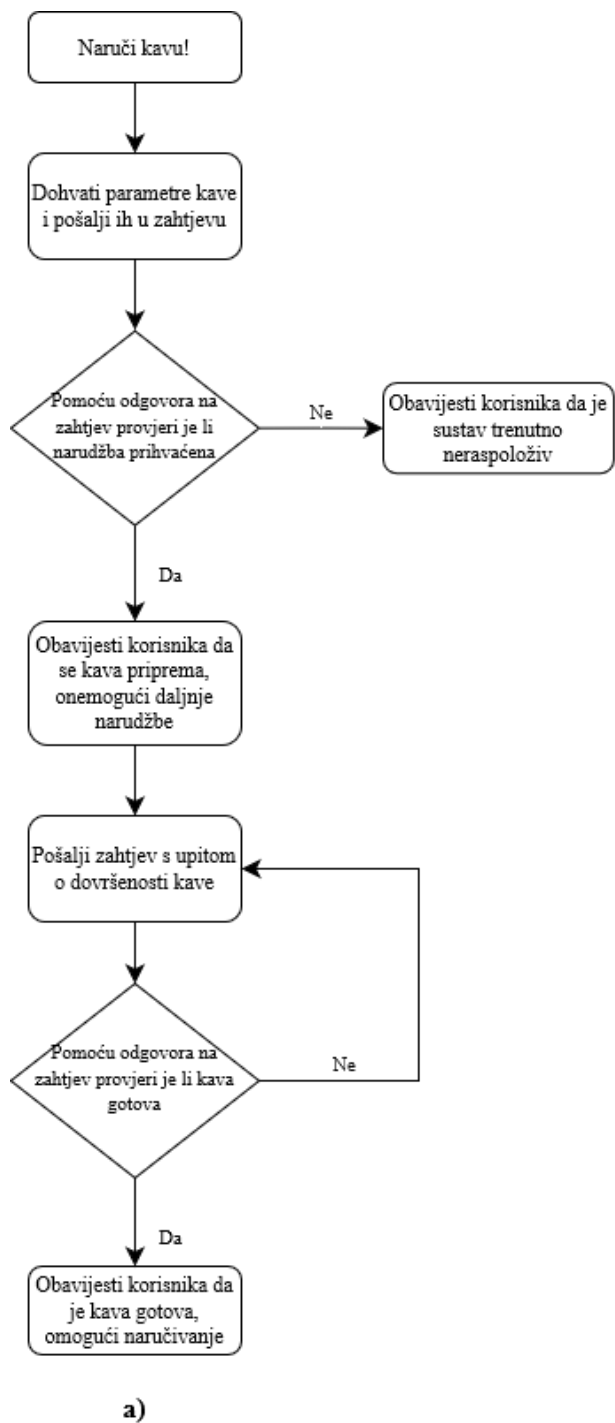


Slika 3.18. Prikaz odabira parametara kave u mobilnoj aplikaciji

Nadalje, slika, opisuje dijagram toka algoritma mobilne aplikacije. Od trenutka kada se mobilna aplikacija pokrene neprestano se provjerava stanje *slidera* za odabir parametara kave, slika b). Ukoliko dođe do promjene nekog od parametara nova vrijednost se sprema u globalnu varijablu za pojedini parametar.

U trenutku kada korisnik pritisne gumb za naručivanje kave izvodi se algoritam prikazan na slici a). Pomoću HTTP Get zahtjeva parametri naručene kave šalju se na Wi-Fi modul koji na zahtjev odgovara sa porukom sadržaja koja otkriva je li narudžba prihvaćena ili odbijena. Ukoliko je

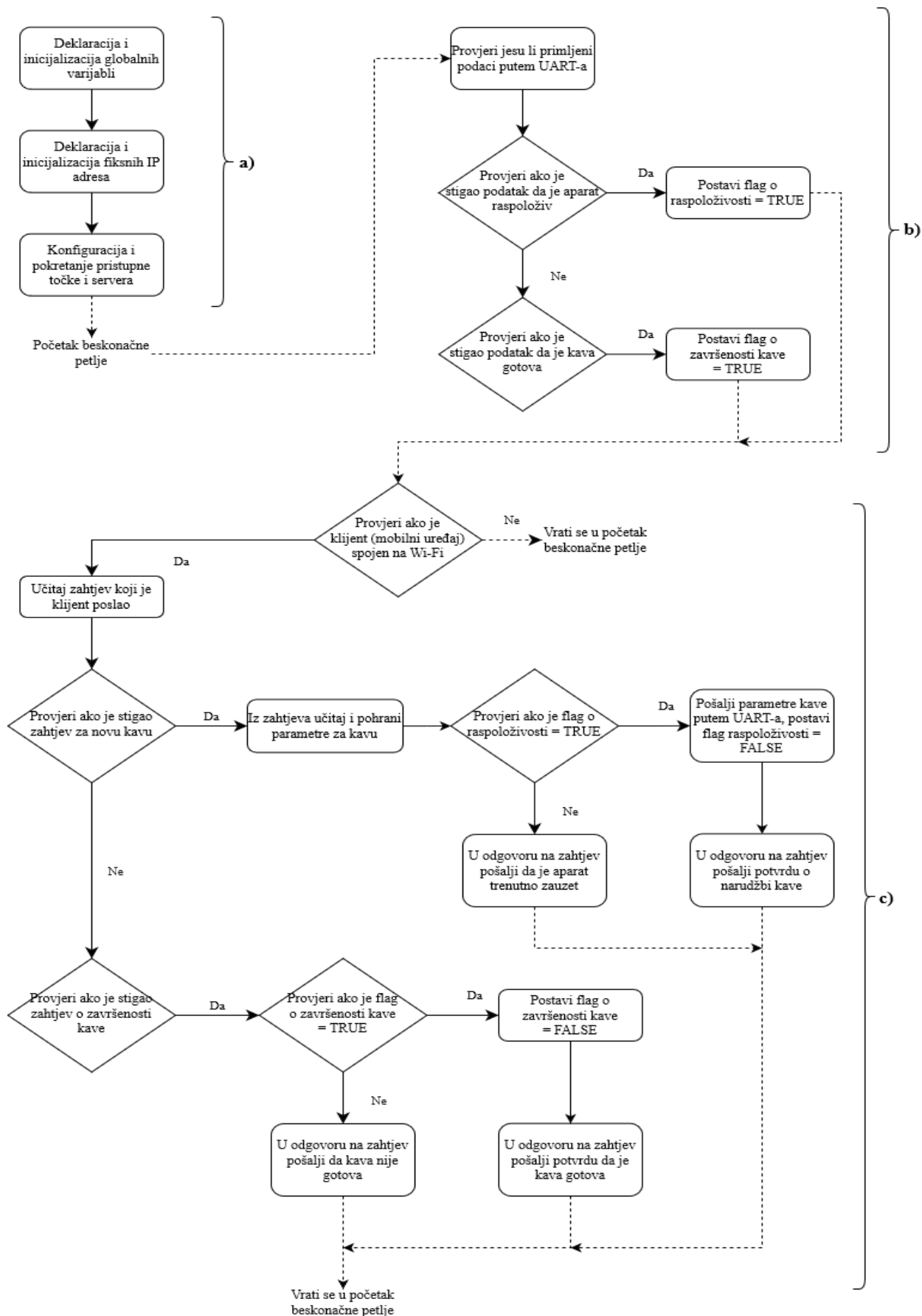
narudžba prihvaćena daljnje naručivanje se onemogućuje sve dok sa STM-a putem Wi-Fi modula ne stigne obavijest da je kava gotova. Svakih 500 ms mobilna aplikacija šalje HTTP Get zahtjev kako bi se iz sadržaja unutar odgovora sa Wi-Fi modula očitalo je li kava gotova ili ne. Nakon što je kava gotova korisnik dobiva obavijest i daljnje narudžbe su omogućene.



Slika 3.19. Dijagram toka algoritma za mobilnu aplikaciju

3.3.3. ESP8266 Wi-Fi modul

Dijagram toka algoritma, slika 3.20., detaljno opisuje ulogu Wi-Fi modula kao posrednika u komunikaciji između mobilnog uređaja i STM32 pločice. Slika 3.20., a), opisuje dio programa koji se izvodi samo na početku, a to je inicijalizacija i deklaracija globalnih varijabli i IP adresa te konfiguracija pristupne točke. Svi flagovi koji se koriste kasnije moraju inicijalno biti postavljeni u stanje FALSE, a pristupna točka konfigurirana je uvijek istom IP adresom. Na slici 3.20., pod b) prikazano je primanje podataka putem UART-a sa STM-a. Svaki put kada se beskonačna petlja pokrene provjerava se je li pristigao novi podatak. Ako je podatak primljen provjerava se njegovo semantičko značenje odnosno je li sustav za pripremu kave spreman za rad ili je gotov sa narudžbom koju trenutno obrađuje. Dio pod c) sa slike 3.20. prikazuje komunikaciju sa mobilnim uređajem. Prvo se provjerava je li uređaj spojen, ukoliko je spojen učitava se zahtjev koji uređaj šalje. Ako je stigao zahtjev za pripremu kave provjerava se je li sustav raspoloživ, ako je raspoloživ parametri za novu kavu koji su primljeni u zahtjevu prosljeđuju se na STM32 pločicu putem UART-a, a mobilnom uređaju u odgovoru na zahtjev stiže potvrda o narudžbi. Ako sustav nije raspoloživ mobilni uređaj obavještava se o neraspoloživosti unutar odgovora na zahtjev. Ako je pristigao zahtjev za provjeru je li trenutna narudžba gotova provjerava se status o dovršenosti primljen putem UART-a, ukoliko je narudžba dovršena u odgovoru na zahtjev se šalje informacija da je kava gotova.



Slika 3.20. Dijagram toka algoritma za mobilnu aplikaciju

4. TESTIRANJE I REZULTATI

4.1. Metode i načini testiranja

Testiranje je važna komponenta svakog proizvoda. Svrha testiranja kod aparata za kavu jest pravilno doziranje kave, šećera, mlijeka i vode kako bi korisnik bio zadovoljan sa okusom te kako ne bi postojali veliki troškovi punjenja samog aparata. Također je bitno da se korisnika ne zakine za količinu kave, norma mora biti ispoštovana, ali s druge strane ukoliko sustav troši više resursa od očekivanog tvrtka je na gubitku. Isto tako pumpa za vodu aparata za kavu mora zadovoljavati normu u količini vode koju treba nasipati, ali mora biti osigurano da se voda neće prelijevati iz šalice i tako trošiti više vode od očekivanog te uz to pružati kavu niže kvalitete. Ideja testiranja je dvadesetak puta poslužiti i izmjeriti količinu kave, šećera, mlijeka u prahu i vode, navedeno će se učiniti za sve kombinacije koje je moguće naručiti u ovom sustavu za pripremu kave. Na taj način dat će se uvid o tome koliko je aparat precizan.

Za dvadesetak mjerenja doza koje je poslužio sustav za pripremu kave za kavu, šećer i mlijeko računat će se aritmetička srednja vrijednost te njena razlika od ciljne vrijednosti. Za testiranje će se koristiti digitalna vaga koja ima korak od 0,1 gram.

Doziranje vode bit će testirano na način da će se u 20 navrata upaliti pumpa te će se mjeriti koliko vode je pumpa nasipala. Pumpa ima protok od 1,5 litara po minuti, a radi 4,2 sekunde te bi trebala nasipati 120 ml vode u šalicu. Za mjerenje se koristi mjerica za tekućinu u mililitrima.

4.2. Rezultati testiranja

Tablice prikazane u ovom poglavlju rezultat su testiranja doziranja aparata za kavu. Na kraju poglavlja analizirani su rezultati i dani su zaključci o uspješnosti izvedbe uređaja.

Tablica 4-1: Testiranje doziranja kave

20 mjerenja u gramima(g)	jedna doza	dvije doze	tri doze
prvo mjerenje	1.5 g	2.9 g	4.2 g
drugo mjerenje	1.5 g	2.8 g	4.1 g
treće mjerenje	1.6 g	2.8 g	4.5 g
četvrto mjerenje	1.7 g	3.0 g	4.5 g
peto mjerenje	1.7 g	2.8 g	4.3 g
šesto mjerenje	1.4 g	3.0 g	4.5 g

sedmo mjerenje	1.4 g	3.0 g	4.5 g
osmo mjerenje	1.6 g	2.7 g	4.1 g
deveto mjerenje	1.4 g	2.7 g	4.5 g
deseto mjerenje	1.7 g	2.7 g	4.1 g
jedanaesto mjerenje	1.6 g	2.8 g	4.3 g
dvanaesto mjerenje	1.6 g	3.1 g	4.3 g
trinaesto mjerenje	1.6 g	3.1 g	4.5 g
četrnaesto mjerenje	1.4 g	3.0 g	4.3 g
petnaesto mjerenje	1.6 g	2.9 g	4.4 g
šesnaesto mjerenje	1.6 g	3.0 g	4.5 g
sedamnaesto mjerenje	1.6 g	3.0 g	4.1 g
osamnajsto mjerenje	1.5 g	3.1 g	4.3 g
devetnajsto mjerenje	1.5 g	3.0 g	4.5 g
dvadeseto mjerenje	1.4 g	3.1 g	4.2 g
srednja vrijednost	1.545 g	2.925 g	4.335 g
ciljana vrijednost	1.5 g	3.0 g	4.5 g
srednja pogreška	0.055 g	0.075 g	0.165 g
maksimalna pogreška	0.2 g	0.3 g	0.4 g

Tablica 4-2: Testiranje doziranja šećera

20 mjerenja u gramima(g)	jedna doza	dvije doze
prvo mjerenje	2.0 g	3.7 g
drugo mjerenje	1.9 g	3.8 g
treće mjerenje	2.0 g	3.6 g
četvrto mjerenje	1.8 g	4.0 g
peto mjerenje	2.2 g	3.6 g
šesto mjerenje	2.1 g	3.7 g
sedmo mjerenje	1.8 g	3.9 g
osmo mjerenje	2.1 g	3.7 g
deveto mjerenje	1.9 g	3.7 g
deseto mjerenje	1.9 g	3.8 g
jedanaesto mjerenje	2.0 g	3.7 g
dvanaesto mjerenje	1.8 g	3.6 g
trinaesto mjerenje	1.6 g	3.8 g
četрнаesto mjerenje	1.8 g	3.9 g
petnaesto mjerenje	1.7 g	4.0 g
šesnaesto mjerenje	2.0 g	3.6 g
sedamnaesto mjerenje	1.9 g	3.8 g
osamnajsto mjerenje	1.8 g	3.7 g
devetnajsto mjerenje	2.1 g	3.7 g
dvadeseto mjerenje	1.9 g	3.8 g
srednja vrijednost	1.915 g	3.94 g
ciljana vrijednost	2.0 g	4.0 g
srednja pogreška	0.085 g	0.06g
maksimalna pogreška	0.4 g	0.4 g

Tablica 4-3: Testiranje doziranja mlijeka

20 mjerenja u gramima(g)	jedna doza	dvije doze
prvo mjerenje	2.2 g	3.7 g
drugo mjerenje	1.8 g	3.8 g
treće mjerenje	2.1 g	3.6 g
četvrto mjerenje	2.2 g	4.0 g
peto mjerenje	2.0 g	3.6 g
šesto mjerenje	2.0 g	3.7 g
sedmo mjerenje	1.8 g	3.9 g
osmo mjerenje	2.1 g	3.7 g
deveto mjerenje	2.0 g	3.7 g
deseto mjerenje	2.2 g	3.8 g
jedanaesto mjerenje	2.1 g	3.7 g
dvanaesto mjerenje	1.9 g	3.6 g
trinaesto mjerenje	2.0 g	3.8 g
četрнаesto mjerenje	2.1 g	3.9 g
petnaesto mjerenje	2.0 g	4.0 g
šesnaesto mjerenje	1.9 g	3.6 g
sedamnaesto mjerenje	2.2 g	3.8 g
osamnajsto mjerenje	2.1 g	3.7 g
devetnajsto mjerenje	2.1 g	3.7 g
dvadeseto mjerenje	2.0 g	3.8 g
srednja vrijednost	2.04 g	3.755 g
ciljana vrijednost	2.0 g	4.0 g
srednja pogreška	0.06 g	0.245 g
maksimalna pogreška	0.2 g	0.4 g

Tablica 4-4: Testiranje doziranja vode

20 mjerenja u mililitrima(ml)	Pumpa radi 4.2 sekunde
prvo mjerenje	112 ml
drugo mjerenje	120 ml
treće mjerenje	119 ml
četvrto mjerenje	119 ml
peto mjerenje	120 ml
šesto mjerenje	117 ml
sedmo mjerenje	118 ml
osmo mjerenje	118 ml
deveto mjerenje	117 ml
deseto mjerenje	120 ml
jedanaesto mjerenje	117 ml
dvanaesto mjerenje	119 ml
trinaesto mjerenje	119 ml
četnaesto mjerenje	118 ml
petnaesto mjerenje	117 ml
šesnaesto mjerenje	119 ml
sedamnaesto mjerenje	120 ml
osamnajsto mjerenje	119 ml
devetnajsto mjerenje	117 ml
dvadeseto mjerenje	117 ml
srednja vrijednost	118.1
ciljana vrijednost	120 ml
srednja pogreška	1.9 ml
maksimalna pogreška	8 ml

Tablica 4-1 prikazuje rezultate testiranja doziranja kave. Iz mjerenja je vidljivo kako je doziranje kave vrlo precizno. U dvadeset mjerenja maksimalno odstupanje od ciljne vrijednosti za jednu dozu kave je samo 0.2 grama što je odstupanje od samo 13.3%. Maksimalno odstupanje od ciljne vrijednosti za dvije doze iznosi 0.3 grama, odnosno 10%, a maksimalno odstupanje za tri doze iznosi 0.4 grama, odnosno 8.8%. Uzimajući u obzir da se mjere vrlo male doze i da se koristi vaga kojoj je korak od 0.1 gram rezultati testiranja su zadovoljavajući.

Tablica 4-2 prikazuje rezultate testiranja doziranja šećera. Iz mjerenja je vidimo kako je doziranje šećera ima veća maksimalna odstupanja od doziranja kave. Kod jedne doze šećera maksimalna pogreška iznosi 0.4 grama, odnosno 20%, što je i dalje zadovoljavajuće. Kod dvije doze maksimalna pogreška se smanjuje na samo 10%. Za šećer je jako bitno naglasiti da se spremnik ne smije nalaziti na izravnom suncu ili u neposrednoj blizini izvora topline jer bi se šećer mogao početi topiti pa bi pogreške kod doziranja bile veće.

Tablica 4-3 prikazuje rezultate testiranje doziranja mlijeka u prahu. iz mjerenja se vidi kako je doziranje mlijeka iznimno precizno. Maksimalna pogreška je 10% i u slučaju doziranja jednom i dvije doze. Mjerenje je konzistentno iz razloga što mlijeko ima postojanu građu za razliku od kave i šećera, ne ljepi se, ne gruš se, konstantno curi jednolikom brzinom.

Tablica 4-4 prikazuje rezultate testiranje doziranja vode. Iz tablice vidimo kako maksimalna pogreška iznosi 8 ml, no ta pogreška je samo prisutna kod prvog doziranja. Razlog tome jest što su cijevi kroz koje ide voda, kod prvog pokretanja, bile u potpunosti prazne. Kod ostalih mjerenja pogreška je bila maksimalno 3 ml, odnosno 2.5%. Pumpa niti jednom nije prešla dozu od 120 ml, iz toga se može zaključiti da neće doći do prelijevanja vode iz šalice.

5. ZAKLJUČAK

Projektni zadatak je uspješno završen. STM32 može bez poteškoća komunicirati sa mobilnom aplikacijom. Kada korisnik pošalje uređaju parametre o željenoj kavi pomoću aplikacije, kava, šećer i mlijeko u prahu se doziraju u šalicu. Zatim grijač grije vodu koju pumpa ulijeva u šalicu kada dođe do zadane temperature. Doziranje se precizno izvodi bez obzira kakve parametre korisnik zada u aplikaciji. Pomoću programskog rješenja, temperaturnog senzora i releja osigurano je da se voda zagrijava do 80°C, a pumpom i dodatnim relejom osigurano je da uređaj ne ulijeva više od 120 mililitara vode u šalicu. Sustav bi mogao biti poboljšán dodavanjem mogućnosti naručivanja više različitih vrsta kave. Također, još jedno moguće unaprjeđenje ovog sustava za pripremu kave bila bi nadogradnja mobilne aplikacije za upravljanje sustavom za pripremu kave koja omogućava korisniku da unese u koje vrijeme želi da mu kava bude pripremljena, na primjer da aplikacija bude povezana sa alarmom na mobitelu.

6. LITERATURA

- [1] Laboratorijske vježbe iz kolegija Ugradbeni računalni sustavi
- [2] STM32F4-Discovery presentation, Nicot31 (18.9.2021.), <http://nicot31.fr/en/stm32f4-discovery-presentation/>
- [3] AI-Thinker ESP8266 ESP-01 Wifi microcontroller, SOLARBOTICS, (19.9.2021), <https://solarbotics.com/product/29246/>
- [4] ESP8266, Wikipedia, (19.9.2021), <https://en.wikipedia.org/wiki/ESP8266>
- [5] Arduino Software (IDE), Arduino, (19.9.2021), <https://www.arduino.cc/en/guide/environment>
- [6] App Inventor for Android, Wikipedia, (19.9.2021), https://en.wikipedia.org/wiki/App_Inventor_for_Android
- [7] Servo motor Towerpro MG995, e-radionica.com, (21.9.2021.), <https://e-radionica.com/hr/servo-motor-towerpro-mg995-360.html>
- [8] DS18B20, Datasheets, (21.9.2021.), <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf?fbclid=IwAR2jm7QejcmZRdN3y2V2h4eQxupSNdnHqTTWjGESVu06D-aJl3vige-N1-I>
- [9] DS18B20, e-radionica.com, (21.9.2021.), https://e-radionica.com/wp/hrvatski/wp-content/uploads/sites/3/2018/03/dsc_8519_1.jpg
- [10] Interface Two Channel Relay Module with Arduino, lastminuteengineers.com, (21.9.2021.), https://lastminuteengineers.com/two-channel-relay-module-arduino-tutorial/?fbclid=IwAR2sX1JzRX9XQnhe9VkrMfgs58SMpGEILa0WwIUztQU_FWM-f7L5r21qxvY
- [11] 2-kanalni relej modul, e-radionica.com, (21.9.2021.), <https://e-radionica.com/hr/modul-s-relejom-2-kanalni.html>
- [12] Bi-Directional Logic Level Converter Hookup Guide, sparkfun.com, (21.9.2021.), https://learn.sparkfun.com/tutorials/bi-directional-logic-level-converter-hookup-guide/all?fbclid=IwAR3BEkkN_6Xe47UUhVBMFAxVAaHYOEKEBars9H7pGTz1KFIR7bzUalT39H8
- [13] Logic level konverter , e-radionica.com, (21.9.2021.), <https://e-radionica.com/hr/logic-level-konverter-eradionica.html>
- [14] Pump, amazon.com, (21.9.2021.), <https://www.amazon.com/Temperature-Resistance-Degrees-Celsius-Diaphragm/dp/B07P1B95S3?fbclid=IwAR1q1v-u7oKAZAsq2CG6GIM-9RI5iW3HIud-KQxe4fxeja95UG0XJ4S3gow>

[15] Library 12- OneWire library for STM32F4, STM32F4 Discovery, (18.9.2021), <http://stm32f4-discovery.net/2014/05/library-12-onewire-library-for-stm43f4xx/>

[16] Library 13- Reading temperature with Dallas DS18B20 on STM32F4, STM32F4 Discovery, (18.9.2021), <https://stm32f4-discovery.net/2014/05/13-reading-temperature-with-dallas-ds18b20-on-stm32f429-discovery-board/>

7. PRILOG

7.1. STM32 programski kod

7.1.1. Korištene biblioteke i globalne varijable

```
/* Includes */
#include "stm32f4xx.h"
#include "uart.h"
#include "defines.h"
#include <string.h>
#include <stdio.h>
#include <tm_stm32f4_onewire.h>
#include <tm_stm32f4_ds18b20.h>

/* Private variables */
volatile int mjera_kave = 900; // vrijednosti Delay-a za koji je otvor otvoren
volatile int mjera_secera = 1300;
volatile int mjera_mlijeka = 600;
volatile int mjera_vode = 4200;
```

7.1.2. Main funkcija

```
int main(void)
{
    char narudzba[3];
    TM_OneWire_t OneWire1;
    uint8_t adresa_senzora[8];
    int temperatura_vode;

    SystemInit();
    USART1_Init();
    config_PWM();
    TM_DELAY_Init();
    config_releji();
    config_temperaturni_senzor(&OneWire1, adresa_senzora);

    GPIO_SetBits(GPIOD,GPIO_Pin_8); // postavljanje releja u početni položaj
    GPIO_SetBits(GPIOD,GPIO_Pin_9);

    Delayms(5000); // delay kako se ne bi primila neodređena poruka Wi-fi modula pri
                  // uključivanju aparata za kavu
    oglasi_pripravnost_korisniku();

    /* Infinite loop */
    while (1)
    {
        ucitaj_narudzbu(narudzba);

        for(int i = 0; i < narudzba[0] - 48; i++){
            pusti_kavu();
        }
    }
}
```

```

    }

    for(int i = 0; i < narudzba[1] - 48; i++){
        pusti_secer();
    }

    for(int i = 0; i < narudzba[2] - 48; i++){
        pusti_mlijeko();
    }

    // provjera temperature vode te grijanje vode dok temperatura ne dosegne 80 stupnjeva
    temperatura_vode = Dohvati_temperaturu(&OneWire1, adresa_senzora);

    if(temperatura_vode <= 80){

        ukljuci_grijac();

        while(temperatura_vode <= 80){
            temperatura_vode = Dohvati_temperaturu(&OneWire1, adresa_senzora);
            Delayms(1000);
        }

        iskljuci_grijac();

    }

    natoci_vodu();

    Delayms(7000); // Sigurnosni delay kako bi sva voda izašla iz cijevi

    obavijesti_kava_je_spremna();

    Delayms(1000);

    oglasi_pripravnost_korisniku();

}
}

```

7.1.3. Ostale korištene funkcije

```

// ----- funkcije vezane za komunikaciju putem UART-a -----
//

void ucitaj_narudzbu(char* narudzba){ // učitavanje narudžbe putem UART-a, narudžba se sprema u
                                     // char polje narudžba

    char znak;
    int counter = 0;

    while(1){
        znak = USART_GetChar();          // primi znak
    }
}

```

```

        if(znak > 47 && znak < 58){           // provjeri je li broj
            narudzba[counter] = znak;        // spremi broj u polje od 3 elementa
            counter++;

            if(counter == 3){
                return;
            }
        }
    }
}

void oglasi_pripravnost_korisniku(){

    USART_PutChar('9');
    USART_PutString("\n");

}

void obavijesti_kava_je_spremna(){

    USART_PutChar('7');
    USART_PutString("\n");

}

// ----- funkcije vezane za digitalne izlaze za releje -----
//

void config_releji(void){

    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOD, ENABLE); // uključivanje takta za port D
    GPIO_InitTypeDef GPIO_InitStructure;                  // inicijalizacijska struktura

    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_8 | GPIO_Pin_9; // 8 - pumpa, 9 - grijač
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;          // izlazni pinovi
    GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;         // push-pull konfiguracija
    GPIO_Init(GPIOD, &GPIO_InitStructure);                // zapisivanje u konfiguracijske registre porta D

}

void natoci_vodu(){

    GPIO_ResetBits(GPIOD,GPIO_Pin_8); // uključivanje pumpe za vodu
    Delayms(mjera_vode);
    GPIO_SetBits(GPIOD,GPIO_Pin_8);   // isključivanje pumpe za vodu

}

void ukljuci_grijac(){

    GPIO_ResetBits(GPIOD,GPIO_Pin_9);

}

```

```

void iskljuci_grijac(){

    GPIO_SetBits(GPIOD,GPIO_Pin_9);

}

// ----- funkcije za PWM servo motore -----//

void config_PWM(void){

    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOC, ENABLE); // Uključivanje takta za port C
    GPIO_InitTypeDef GPIO_InitStructurePWM; // Inicijalizacijska struktura za GPIO

    // Inicijalizacija PC6, PC7 i PC8 (TIM3 Ch1, Ch2 i Ch3)
    GPIO_InitStructurePWM.GPIO_Pin = GPIO_Pin_6 | GPIO_Pin_7 | GPIO_Pin_8;
    GPIO_InitStructurePWM.GPIO_Mode = GPIO_Mode_AF;
    GPIO_InitStructurePWM.GPIO_Speed = GPIO_Speed_100MHz;
    GPIO_InitStructurePWM.GPIO_OType = GPIO_OType_PP;
    GPIO_InitStructurePWM.GPIO_PuPd = GPIO_PuPd_UP;
    GPIO_Init(GPIOC, &GPIO_InitStructurePWM);

    GPIO_PinAFConfig(GPIOC, GPIO_PinSource6, GPIO_AF_TIM3); // Alternativna funkcija pina
    //PB6, PB7 i PB8 je PWM signal s TIM3
    GPIO_PinAFConfig(GPIOC, GPIO_PinSource7, GPIO_AF_TIM3);
    GPIO_PinAFConfig(GPIOC, GPIO_PinSource8, GPIO_AF_TIM3);

    RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM3, ENABLE); // Uključivanje takta za TIM3

    TIM_TimeBaseInitTypeDef TIM3_TimeBaseStructure; // Struktura za PWM GPIO

    // Podešavanje TIM3 vremenske baze - frekvencija PWM signala je 50Hz
    TIM3_TimeBaseStructure.TIM_Period = 19999;
    TIM3_TimeBaseStructure.TIM_Prescaler = 84;
    TIM3_TimeBaseStructure.TIM_ClockDivision = TIM_CKD_DIV1;
    TIM3_TimeBaseStructure.TIM_CounterMode = TIM_CounterMode_Up;
    TIM_TimeBaseInit(TIM3, &TIM3_TimeBaseStructure);
    TIM_Cmd(TIM3, ENABLE); // Pokreni timer

    // Podešavanje karakteristika PWM signala
    TIM_OCInitTypeDef TIM_OCInitStructure;

    // Podešavanje
    TIM_OCInitStructure.TIM_OCMode = TIM_OCMode_PWM1;
    TIM_OCInitStructure.TIM_OutputState = TIM_OutputState_Enable;
    TIM_OCInitStructure.TIM_OCPolarity = TIM_OCPolarity_High;
    TIM_OCInitStructure.TIM_Pulse = 0; // početna vrijednost duty cycle-a = 0%

    // PWM signal na kanal 1
    TIM_OC1Init(TIM3, &TIM_OCInitStructure);
    TIM_OC1PreloadConfig(TIM3, TIM_OCPreload_Enable);

    // PWM signal na kanal 2
    TIM_OC2Init(TIM3, &TIM_OCInitStructure);

```



```

TIM_OC2PreloadConfig(TIM3, TIM_OCPreload_Enable);

// PWM signal na kanal 3
TIM_OC3Init(TIM3, &TIM_OCInitStructure);
TIM_OC3PreloadConfig(TIM3, TIM_OCPreload_Enable);
}

void pusti_kavu(){

    TIM3->CCR1 = 1200; // otvori otvor , vrijednosti u mikrosekundama za duty cycle
    Delayms(mjera_kave);
    TIM3->CCR1 = 700; // otvori otvor
    Delayms(mjera_kave);
}

void pusti_secer(){

    TIM3->CCR2 = 1200;
    Delayms(mjera_secera);
    TIM3->CCR2 = 600;
    Delayms(mjera_secera);
}

void pusti_mlijeko(){

    TIM3->CCR3 = 1500;
    Delayms(mjera_mlijeka);
    TIM3->CCR3 = 2200;
    Delayms(mjera_mlijeka);
}

// ----- funkcije za temperaturni senzor -----//

void config_temperaturni_senzor(TM_OneWire_t* OneWire1, uint8_t* adresa_senzora){

    // Inicijalizacija OneWire instance za pin PE15
    TM_OneWire_Init(OneWire1, GPIOE, GPIO_Pin_15);

    uint8_t senzor_je_spojen;

    // provjera ima li spojenih OneWire senzora
    senzor_je_spojen = TM_OneWire_First(OneWire1);
    if(senzor_je_spojen){
        // dohvaćanje 8 bajtna rom adrese spojenog senzora na temelju koje se može
        // manipulirati određenim senzorom
        TM_OneWire_GetFullROM(OneWire1, adresa_senzora);

        // postavljanje 9 bitne rezolucije senzora
        TM_DS18B20_SetResolution(OneWire1, adresa_senzora,
                                TM_DS18B20_Resolution_9bits);
    }
}

```

```

        // onemogućavanje alarma za postizanje određene temperature
        TM_DS18B20_DisableAlarmTemperature(OneWire1, adresa_senzora);
    }
}

int Dohvati_temperaturu(TM_OneWire_t* OneWire1, uint8_t* adresa_senzora){

    float temperatura;
    // početak pretvorbe temperature u svim senzorima koji su spojeni
    // na pin iz OneWire strukture
    TM_DS18B20_StartAll(OneWire1);

    // čekanje dok svi senzori koji su spojeni ne obave pretvorbu temeperature
    while (!TM_DS18B20_AllDone(OneWire1));

    // čitanje temperature sa senzora
    if (TM_DS18B20_Read(OneWire1, adresa_senzora, &temperatura)){
        return (int)temperatura;
    }

    return 0;
}

```

7.2. ESP8266 programski kod

7.2.1. Korištene biblioteke i globalne varijable

```
// includes
#include <ESP8266WiFi.h>

//inicijalizacija TCP servera
WiFiServer server(80);

//inicijalizacija i deklaracija varijabli
String request = "";
int caffeValue = 0;
int flagReady = 0;
int kavaResponse = 0;
int intSerialStm = 0;
String strSerialStm = "";
int flagFirstTime = 0;
int coffeeFinished = 0;
int pump;

//inicijalizacija i deklaracija IP adresa
IPAddress local_IP(192, 168, 4, 1);
IPAddress gateway(192, 168, 1, 1);
IPAddress subnet(255, 255, 0, 0);
```

7.2.2. Setup funkcija

```
void setup()
{
    //određivanje baud-rate UART-a
    Serial.begin(115200);
    Serial.println();

    //konfiguracija WiFi IP adresa
    WiFi.config(local_IP, gateway, subnet)

    //konfiguracija pristupne točke
    WiFi.softAP("CoffeeMaker", "12345678");

    //pokretanje servera
    server.begin();
}
```

7.2.3. Loop funkcija

```
void loop()
{
    //prvi puta program ne učitava ništa putem UART-a 3s
    //kako ne bi pokupio "garbage" pri pokretanju STM-a
    if(flagFirstTime == 0){
        delay(3000);
        flagFirstTime = 1;
    }

    //očitanje podataka primljenih putem UART-a
    if(Serial.available()){
        strSerialStm = Serial.readString();
        intSerialStm = strSerialStm.toInt();
    }

    //provjera podatka o raspoloživosti aparata za kavu
    if(intSerialStm == 9){
        flagReady = 1;
        intSerialStm = 0;
    }

    //provjera podatka o tome je li kava gotova
    if(intSerialStm == 7){
        coffeeFinished = 1;
        intSerialStm = 0;
    }

    //čekanje klijenta da se prispoji
    WiFiClient client = server.available();
    if(!client){
        return;
    }

    //očitanje zahtjeva kojega je klijent poslao
    request = client.readStringUntil('\r');

    //provjera je li stigao zahtjev za novu kavu
    if(request.indexOf("kava") > 0){
        caffeValue = request.substring(10, 13).toInt();

        //ako je aparat za kavu raspoloživ putem UART-a se naručuje nova kava
        if(flagReady == 1){
            Serial.println(caffeValue);
            flagReady = 0;
            kavaResponse = 1;
        }
        else{
            kavaResponse = 0;
        }
    }
}
```

```

    //odgovor na zahtjev klijenta
    client.flush();

    client.println("HTTP/1.1 200 OK");
    client.println("Content-type:text/html");
    client.println();
    //sadržaj odgovora ukazuje na to je li kava naručena ili je
    //aparati za kavu trenutno zauzeti
    client.println(String(kavaResponse));
    client.println();

    } //provjera je li stigao zahtjev o završenosti trenutne narudžbe
else if(request.indexOf("gotova") > 0){

    //odgovor na zahtjev klijenta
    client.flush();

    client.println("HTTP/1.1 200 OK");
    client.println("Content-type:text/html");
    client.println();
    //sadržaj odgovora ukazuje na to je li kava gotova
    client.println(String(coffeeFinished));
    client.println();

    if(coffeeFinished == 1){
        coffeeFinished = 0;
    }
    }

    delay(5);
}

```

7.3. Mobilna aplikacija programski blokovi

